*Journal of Engineering Technology*
*Volume 8, Issue 1, Jan. 2019, PP.143-157*
RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

# Improved on an Efficient User Authentication Scheme for Heterogeneous Wireless Sensor Network Tailored for the Internet of Things Environment

**Yalin Chen[1] , Jue-Sam Chou[*2] , Hung - Sheng Wu[3] , Hung-Pin Chiu[4]**

[1]Institute of information systems and applications, National Tsing Hua University
[*2,4] Department of Information Management, Nanhua University, Taiwan
[3] Department of Information Management, Nanhua University, Taiwan
[*2]: corresponding author

**Abstract:** Recently, Farash et al. proposed an efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. By using BAN-logic and AVISPA tools, they confirmed the security properties of the proposed scheme. Yet, after analyzing, we determined that the scheme could not resist the smart card loss password guessing attack and suffers anonymity breach, which are two of the ten basic requirements in a secure identity authentication using smart card, insisted by Liao et al. Thus, we modified their method to include the desired security functionalities. After verification, we confirmed that the modified scheme satisfies the ten needed security attributes, which are important in a user authentication protocol using smart card. Moreover, after comparisons, we found it also is either safer or more efficient with only two passes than several state of the art work.

**Keywords:** user authentication, key agreement, cryptanalysis, smart card, password change, wireless sensor network, Internet of Things, anonymity, hash function

## 1 Introduction

There have been many cryptographic scientists working in the field of unbreakable encryption [21] and thus leads to the design of identity authentication system using smart cards [1-18]. The heterogeneous wireless sensor network identity authentication system [6] is one of such systems, which contain three roles: user, sensor node, and the gateway node (GWN); and three protocols: registration, login and authentication, and password change. In the design principle, the user's identity should not be revealed to ensure his login privacy. In 2016, Farasha et al. [11] point out that they have found some security shortcomings in Turkanovic et al.'s scheme [6], which makes it susceptible to some cryptographic attacks. They hence overcome the weaknesses by proposing a new improved user authentication and key agreement scheme (UAKAS). The proposed scheme enhances the security level and enables the heterogeneous wireless sensor networks (WSN) , which have gradually changed to scalar Multimedia Sensor Networks for user to access video, images, and audio [19], to dynamically grow without influencing any involved party. They claim that the security analysis results,

instructed by using BAN-logic and AVISPA tools, confirms the proposed scheme's security. But, upon a closer examination, we discover that it does not support the needed security when an attacker launches a smart card loss password guessing attack. To overcome this weakness, we modify their scheme to include this feature. We will demonstrate the enhancement in this article. In addition, in 2017 Dhillon et al. [16] propose a protocol, "a lightweight biometrics based remote user authentication scheme for IoT services", and declare that the scheme is robust against multiple security attacks. Nonetheless, we found that from the parameters stored in smartphone memory, like the ones stored in a smart card memory, if an attacker gets a user $U_i$'s lost smartphone memory, he can launch a password guessing attack by computing $y_i=e_i \oplus x_i = H(H(r_i||PW_i)||x_{gu})$, where $r_i$ and $x_{gu}$ are the stored values. Therefore, their scheme suffers the lost smart card password guessing attack. In 2018, Gupta et al. [14] propose a lightweight anonymous user authentication and key establishment scheme for wearable devices, which is a good design; however, we found the scheme needs to store a verifier table on the server side. This violates one of the ten security requirements for an authentication scheme advocated by Liao et al. Besides, the two parameters $MGID_i$, $MSID_i$ keep unchanged forever, which might incur some malicious attempts. Meanwhile, each $GWN_i$ can launch an offline $X_{ser}$ (the server's secret) guessing attack, because $e_i$ equals to $h(MI_u \,||\, X_{ser}) \oplus h(MP_u \,||\, X_{GWNi})$. Also, Sharma et al. [15] propose a lightweight a lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications in 2018. They state that their scheme is robust against attacks. Yet, we found after the registration each user $LU_i$, a legal insider, can calculate out $MV_i=Z_i+X_i$, where $Z_i$ is stored in the smart card and $X_i$ can be computed by himself using $h(username_i||MPW_i)=h(username_i||(R_1||PW_i))$. After that, he can launch an offline server's private key, *V*, guessing attack by computing $MV_i=h(MID_i||V)$. Then, once he has intercepted the other user $U_i$'s $MID_i$, he can impersonate $U_i$ to login to the server at his will. In 2019, Lwamo et al. [17] propose a scheme "a secure user authentication scheme with anonymity for the single and multi-server environments", which they claim is reliable through mutual authentication and resilient to malicious attacks. However, we discovered a defect in their design that when the server receives the login message from the user, he has no idea about who is the user and thus cannot use the corresponding $K_{mx}$ to decrypt $ID_{im}$ for getting $ID_{ih}$.

The rest of this paper is organized as follows. Section 2 review Farasha et al.'s scheme. Section 3 presents the weaknesses of their scheme. Section 4 describes our modifications in the registration phase, and the login and authentication phase. Section 5 analyzes the security of the modification. Then, we make comparisons among our scheme with some others in the state of the art in Section 6. Finally, a conclusion is given in Section 7.

## 2 Reviews of Farash et al.'s scheme

Farash et al.'s heterogeneous wireless sensor network identity authentication for the Internet of Things [20] is based on Turkanovic et al.'s scheme [6]. It consists of three roles: users, sensor nodes, and a gateway node (GWN); and some phases: pre-deployment, registration, login and authentication, password change, and dynamic node addition phase. They claimed that their method, not only eliminates all security vulnerabilities existing in Turkanovic et al.'s scheme, but also enhance its security level, which enables the WSN's unlimitedly grow and makes the functionality and efficiency reach the same level as theirs. In this article, we only review the registration phase, and login and

authentication phase to illustrate the weaknesses. As for the used notations' definitions, please refer to the original article.

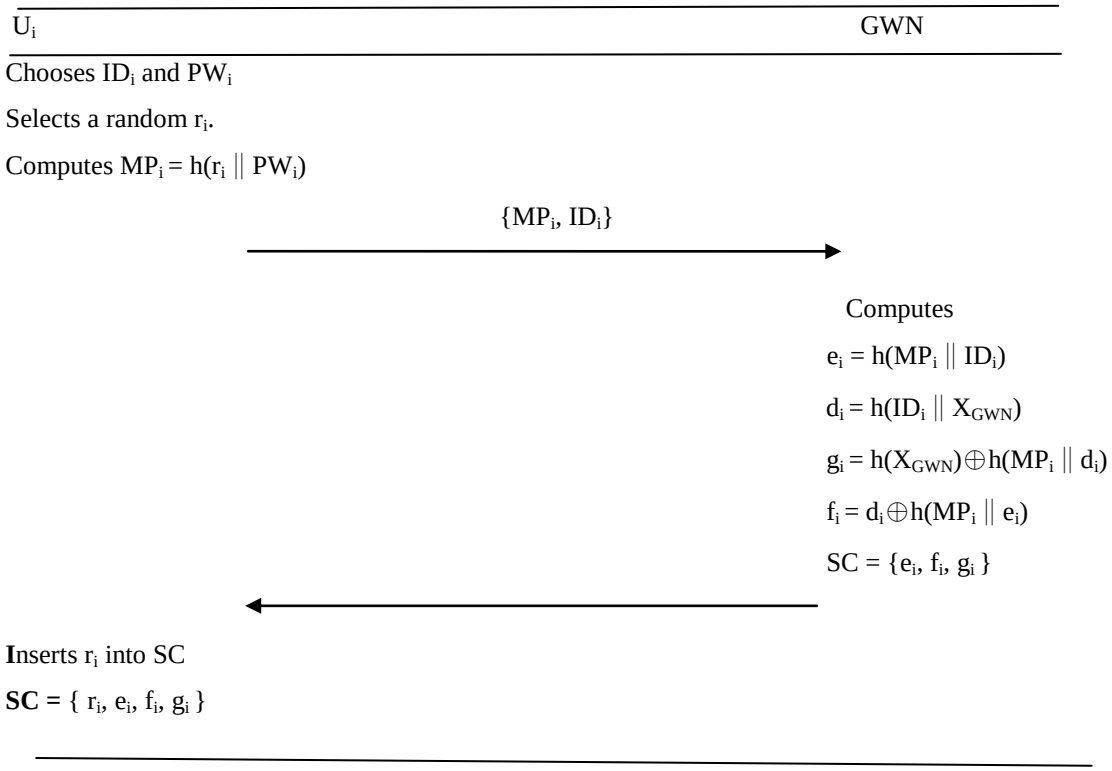
### 2.1 Registration Phase

This phase is divided into two parts: (a) the user registration phase, and (b) the sensor node registration phase. We describe both below and depict them in Fig 1 and 2 respectively.

**(a). The user registration phase**

As shown in Fig 1, the user $U_i$ chooses his username $ID_i$, password $PW_i$, and selects a random nonce $r_i$. He then computes $MP_i = h(r_i \| PW_i)$ and sends $\{MP_i, ID_i\}$ to GWN over a secure channel. After receiving the registration message from $U_i$, GWN first computes value $e_i = h(MP_i \| ID_i)$, then computes $d_i = h(ID_i \| X_{GWN})$, $g_i = h(X_{GWN}) \oplus h(MP_i \| d_i)$, and $f_i = d_i \oplus h(MP_i \| e_i)$ by using $U_i$'s secret combined with its secret master key $X_{GWN}$. It stores $\{e_i, f_i, g_i\}$ into the smart card (SC) and sends SC to $U_i$. After receiving SC, $U_i$ inserts $r_i$ into it, and terminates the registration phase.

**Fig. 1.** user registration phase of Farash's scheme




**(b). The sensor node registration phase**

A specific sensor $S_j$ has to register to the GWN with a message $\{SID_j, MP_j, MN_j, T_1\}$ over an insecure

channel. This message is generated by $S_j$, which first randomly selects a nonce $r_j$, then computs $MP_j = h(X_{GWN-Sj} \| r_j \| SID_j \| T_1)$ and $MN_j = r_j \oplus X_{GWN-Sj}$. After receiving the registration message from $S_j$, GWN checks whether $|T_1 - T_c| < \triangle T$ holds, if the verification holds, GWN computes the random nonce $r_j' = MN_j \oplus X_{GWN-Sj}$ and $MP_j' = h(X_{GWN-Sj} \| r_j' \| SID_j \| T_1)$, and checks to see if it is equal to the received $MP_j$. If it is, GWN computes the values $x_j = h(SID_j \| X_{GWN})$, $e_j = x_j \oplus X_{GWN-Sj}$, $d_j = h(X_{GWN} \| 1) \oplus h(X_{GWN-Sj} \| T_2)$, and $f_j = h(x_j \| d_j \| X_{GWN-Sj} \| T_2)$. Then sends $S_j$ the following message $\{e_j, f_j, d_j, T_2\}$. $S_j$ then checks whether $|T_2 - T_c| < \triangle T$. If the verification holds, $S_j$ computes $x_j = e_j \oplus X_{GWN-Sj}$ and compares $f_j$ with $h(x_j \| d_j \| X_{GWN-Sj} \| T_2)$. If they are equal, $S_j$ calculates $h(X_{GWN} \| 1) = d_j \oplus h(X_{GWN-Sj} \| T_2)$ and stores $h(X_{GWN} \| 1)$ and $x_j$ into its memory. Finally, $S_j$ deletes $X_{GWN-Sj}$ and $SID_j$, and sends a confirmation message to GWN.

**Fig. 2.** Sensor node registration phase of Farash's scheme

| $S_j$ | GWN |
|---|---|
| | Knows its master key $X_{GWN}$ |
| Stores its $SID_j$ and $X_{GWN-Sj}$ | For each $S_j$ stores their $SID_j$ and $X_{GWN-Sj}$ |
| Selects a random $r_j$ | |
| Computes | |
| $MP_j = h(X_{GWN-Sj} \| r_j \| SID_j \| T_1)$ | |
| $MN_j = r_j \oplus X_{GWN-Sj}$ | |
| | $\{SID_j, MP_j, MN_j, T_1\}$ → |
| | Checks $\|T_1 - T_c\| < \triangle T$ |
| | Computes |
| | $r_j' = MN_j \oplus X_{GWN-Sj}$ |
| | $MP_j' = ? h(X_{GWN-Sj} \| r_j' \| SID_j \| T_1)$ |
| | $x_j = h(SID_j \| X_{GWN})$ |
| | $e_j = x_j \oplus X_{GWN-Sj}$ |
| | $d_j = h(X_{GWN} \| 1) \oplus h(X_{GWN-Sj} \| T_2)$ |
| | $f_j = h(x_j \| d_j \| X_{GWN-Sj} \| T_2)$ |
| | ← $\{e_j, f_j, d_j, T_2\}$ |
| Checks $\|T_2 - T_c\| < \triangle T$ | |
| Computes | |
| $x_j = e_j \oplus X_{GWN-Sj}$ | |
| $f_j = ? h(x_j \| d_j \| X_{GWN-Sj} \| T_2)$ | |
| $h(X_{GWN} \| 1) = d_j \oplus h(X_{GWN-Sj} \| T_2)$ | |
| Stores $x_j$ and $h(X_{GWN} \| 1)$ into a memory | |

Deletes $X_{GWN-S\,j}$ and $SID_j$ from memory

confirmation →

Deletes $SID_j$ and $X_{GWN-S\,j}$ from memory

## 2.2 Login and authentication phase

This phase enables a user to negotiate a session key with a specific sensor node without contacting the GWN. The session key will be used later for secure communication between the user and the sensor node.

**Fig. 3.** Loginin and authentication phase of Farash's scheme

| $U_i$ | $S_j$ | GWN |
|---|---|---|
| Knows its $ID_i$ , $PW_i$ | Stores $SID_j$ , $x_j$ and $h(X_{GWN} \| 1)$ | Stores its master key $X_{GWN}$ |
| Has a SC = { $r_i$, $e_i$, $f_i$, $g_i$ } | | |

User

Inserts SC into a terminal

Inputs $ID_i'$ and $PW_i'$

SC computes

$MP_i' = h(r_i \| PW_i')$

$e_i = ?\ h(MP_i' \| ID_i')$

$d_i = f_i \oplus h(MP_i' \| e_i)$

$h(X_{GWN}) = g_i \oplus h(MP_i' \| d_i)$

$M_1 = ID_i' \oplus h(h(X_{GWN}) \| T_1)$

Chooses a random nonce $K_i$

$M_2 = K_i \oplus h(d_i \| T_1)$

$M_3 = h(M_1 \| M_2 \| K_i \| T_1)$

Chooses $S_j$

$\{M_1, M_2, M_3, T_1\}$ →

Checks $|T1 - Tc| < \triangle T$

$ESID_j = SID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$

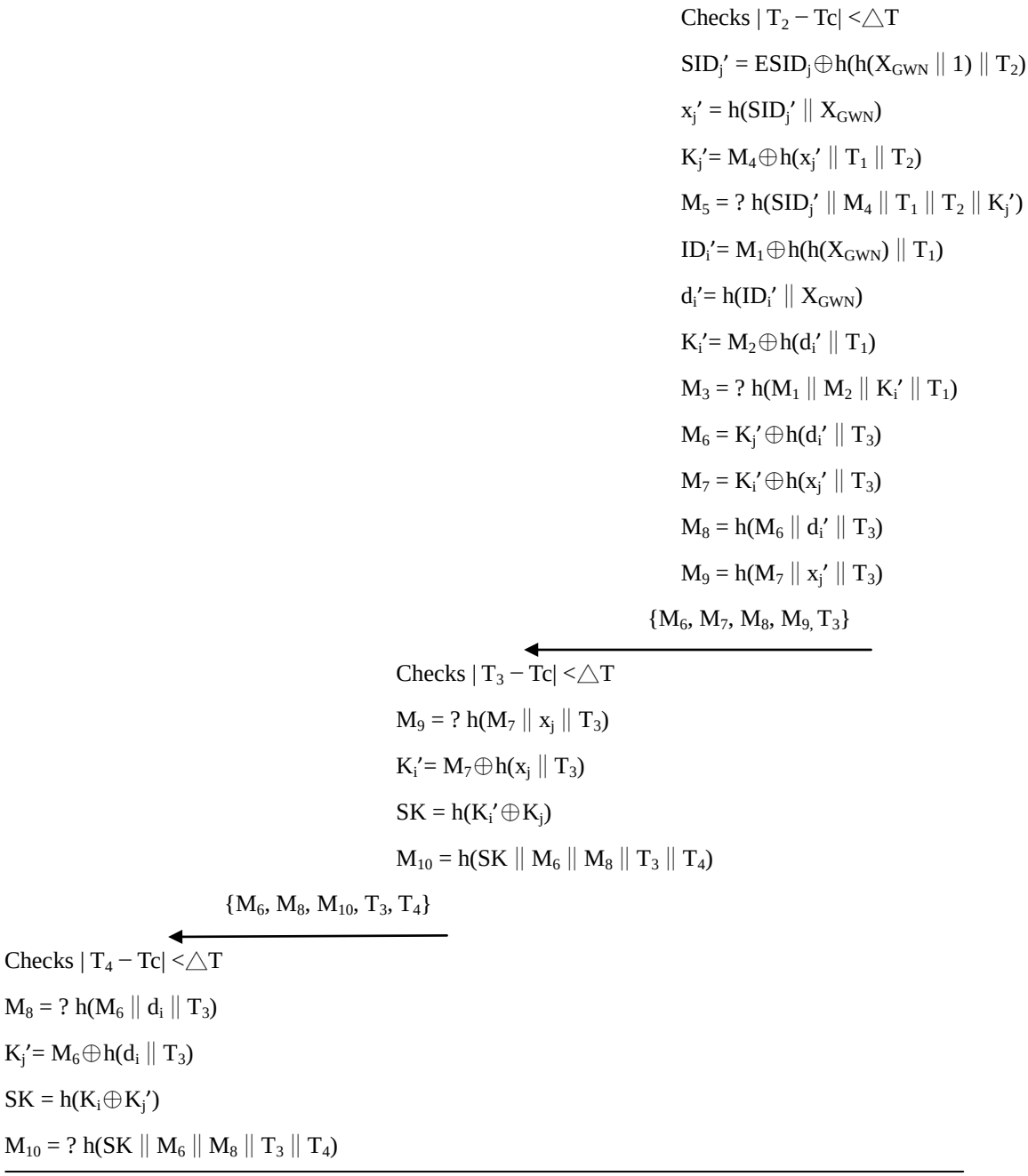
Chooses a random nonce $K_j$

$M_4 = h(x_j \| T_1 \| T_2) \oplus K_j$

$M_5 = h(SID_j \| M_4 \| T_1 \| T_2 \| K_j)$

$\{M_1, M_2, M_3, T_1, T_2, ESID_j, M_4, M_5\}$ →

*Journal of Engineering Technology*
*Volume 8, Issue 1, Jan. 2019, PP.143-157*
RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY




**(a). Login phase**

Ui inserts his SC into a card reader, and inputs its username $ID_i$ and password $PW_i$. SC then verifies its owner by using the stored secret $PW_i$ and $r_i$. First, it computes $MP_i = h(r_i || PW_i)$, then $e_i' = h(MP_i || ID_i)$, and compares $e_i'$ with the stored $e_i$ to see if they are equal. If they are, SC confirms the legitimacy of $U_i$.

**(b). Authentication phase**

SC first computes $d_i = f_i \oplus h(MP_i \| e_i)$, by using the stored values $f_i$, $e_i$, and the $MP_i$ from login phase. It then computes $h(X_{GWN}) = g_i \oplus h(MP_i \| d_i)$, where gi is stored in SC. After that, it computes $M_1 = ID_i \oplus h(h(X_{GWN}) \| T_1)$ and randomly chooses a secret nonce $K_i$ to calculate $M_2 = K_i \oplus h(d_i \| T_1)$, where $T_1$ is SC's current timestamp. Finally, SC computes $M_3 = h(M_1 \| M_2 \| K_i \| T_1)$ and sends the authentication message $\{M_1, M_2, M_3, T_1\}$ to the sensor node $S_j$ via an insecure channel. After receiving the message from $U_i$, $S_j$ first checks to see whether ($|T_1 - T_c| < \Delta T$) holds, where Tc is $S_j$'s current timestamp. If it holds, $S_j$ computes $ESID_j = SID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$ and randomly chooses a nonce $K_j$ to compute the value $M_4 = h(x_j \| T_1 \| T_2) \oplus K_j$, where xj is the stored value, $T_1$ is $U_i$'s initial timestamp, and $T_2$ $S_j$'s current timestamp. $S_j$ then uses value $M_4$, its identity $SID_j$, $K_j$, and the timestamps to compute $M_5 = h(SID_j \| M_4 \| T_1 \| T_2 \| K_j)$, and then sends message $\{M_1, M_2, M_3, T_1, T_2, ESID_j, M_4, M_5\}$ to GWN. After receiving the message from $S_j$, GWN first checks for a replay attack. If it does not happen, GWN computes $S_j$'s identity $SID_j = ESID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$, by using $ESID_j$ and $T_2$ both received in the message, alongside with its own secret master key $X_{GWN}$. After that, GWN computes $x_j = h(SID_j \| X_{GWN})$ and $Kj' = M_4 \oplus h(x_j \| T_1 \| T_2)$, and verifies the legitimacy of $S_j$ by computing $M_5' = h(SID_j' \| M_4 \| T_1 \| T_2 \| K_j')$ and comparing whether $M_5'$ is equal to the received one. If it equals, GWN confirms that $S_j$ is authentic. It then computes $ID_i' = M_1 \oplus h(h(X_{GWN}) \| T_1)$, $d_i' = h(ID_i' \| X_{GWN})$, and $K_i' = M_2 \oplus h(d_i' \| T_1)$, and checks whether the received $M_3$ is equal to $h(M_1 \| M_2 \| K_i' \| T_1)$. If it is, GWN confirms the legitimacy of $U_i$ and prepares four auxiliary values $M_6$, $M_7$, $M_8$ and $M_9$ by computing $M_6 = K_j \oplus h(d_i \| T_3)$, $M_7 = K_i \oplus h(x_j \| T_3)$, $M_8 = h(M_6 \| d_i \| T_3)$, and $M_9 = h(M_7 \| x_j \| T_3)$, respectively. GWN finally sends them to $S_j$. If $S_j$ receives the confirmation message from GWN, it knows that $U_i$ is legitimate and then checks for any replay attack. If it isn't a replay attack, $S_j$ checks the legitimacy of the received message by calculating $M_9 = h(M_7 \| x_j \| T_3)$ and comparing it with the received one. If the verification holds, $S_j$ computes $K_i' = M_7 \oplus h(x_j \| T_3)$ and constructs the session key $SK = h(K_i' \oplus K_j)$. Finally, it computes $M_{10} = h(SK \| M_6 \| M_8 \| T_3 \| T_4)$ and sends $\{M_6, M_8, M_{10}, T_3, T_4\}$ to $U_i$. $U_i$ also checks for any replay attacks and verifies the legitimacy of the received message to avoid any GWN or Sj impersonation attacks. If a replay attack is ruled out, $U_i$ computes the value $M_8 = h(M_6 \| d_i \| T_3)$ and compares it with the received one. If they are equal, it stands for, that $U_i$ successfully verifies GWN. After that, $U_i$ calculates $K_j' = M_6 \oplus h(d_i \| T_3)$ and $SK = h(K_i \oplus K_j')$. And verifies the legitimacy of SK by comparing whether the received $M_{10}$ is equal to $h(SK \| M_6 \| M_8 \| T_3 \| T_4)$. If they are equal, $U_i$ ensures the authenticity of $S_j$.

## 3 Weakness of the scheme

Due to that the smart card stores the parameters $f_i$, $e_i$, $g_i$ ,$r_i$ and the user himself can compute the value $MP_i$, if the user plays the role of an inside attacker, he can compute his own $d_i = f_i \oplus h(MP_i \| e_i)$ and $h(X_{GWN}) = g_i \oplus h(MP_i \| d_i)$. That is, each insider can know the value $h(X_{GWN})$. Under this situation, we can see that their scheme suffers both (1) The smart card loss password guessing attack, and (2) Anonymity breach. We describe them both the reasons why in the following.

### 3.1 The smart card loss password guessing attack

If a user loses his smart card, which is then obtained by an inside attacker, the insider can launch a smart card loss password guessing attack as follows.
The insider first calculates $A=g_i' \oplus h(X_{GWN})$ and guesses the lost card owner's password as pwi'. He then computes $MP_i'=h(r_i' \| pw_i')$, $d_i'= f_i' \oplus h(MP_i' \| e_i')$, and $h(MP_i' \| d_i')$, where $r_i'$, $g_i'$, $f_i'$, $e_i'$ are the parameters stored in the lost smart card. That is, if the attacker guesses the right password $pw_i'$, he will get the user's $d_i'$, then the computed value $h(MP_i' \| d_i')$ will definitely equals to A. So, the attacker can confirm that he succeeds.

### 3.2 Anonymity breach

Due to that $M_1 = ID_i \oplus h(h(X_{GWN}) \| T_1)$ and $ESID_j = SID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$, and both the transferred messages in the login and authentication phase, $\{M_1, M_2, M_3, T_1\}$ from $U_i$ to $S_j$ and $\{M_1, M_2, M_3, T_1, T_2, ESID_j, M_4, M_5\}$ from $S_j$ to GWN, an insider can compute $ID_i= M_1 \oplus h(h(X_{GWN}) \| T1)$ from the calculated $h(X_{GWN})$ and an insider sensor node can compute $SID_j = ESID_j \oplus h(h(X_{GWN} \| 1) \| T_2)$ from the sensor's stored $h(X_{GWN} \| 1)$. Thus, their scheme does not own the anonymous property for both the user and the sensor node.

## 4 Modification

From the weaknesses found in Section 3, we note that the key point is that the insider can obtain GWN's secret $h(X_{GWN})$. Hence, it needs to be disguised. We have thus changed the messages in the registration phase, and the login and authentication phase as follows. We also show the results in Fig4 and 5, respectively.

### 4.1 For user i

First, we modify user i's stored value gi to be $h( h(X_{GWN}) \oplus h(e_i \oplus ID_i \oplus d_i) ) \oplus h(MP_i \| d_i)$, which is set as $h(X_{GWN}) \oplus h(MP_i \| d_i)$. Hence, $h(h(X_{GWN}) \oplus h(e_i \oplus ID_i \oplus d_i)) = g_i \oplus h(MP_i \| d_i)$ in the login and authentication phase at the user side. Let $M12 = h(e_i \oplus ID_i \oplus d_i)$. Then, the user computes $M_1 = ID_i \oplus h( (g_i \oplus h(MP_i \| d_i) ) \| T_1 ) = ID_i \oplus h( h( h(X_{GWN}) \oplus M_{12} ) \| T_1 )$ and transfers the authentication message $\{M_1, M_2, M_3, M_{12}, T_1\}$ to the sensor node $S_j$.

**Fig. 4.** Modified User (Ui) Login and Authentication Phase

| User (Ui) | Sensor Node ($S_j$) |
|---|---|
| **Login and Authentication Phase** | |
| Modify user i's stored value | |

$g_i = h( h(X_{GWN}) \oplus h(e_i \oplus ID_i \oplus d_i) ) \oplus h(MP_i \| d_i)$, which is Originally set as $g_i = h(X_{GWN}) \oplus h(MP_i \| d_i)$

Lets

$M_{12} = h(e_i \oplus ID_i \oplus d_i)$.

Computes

$M_1 = ID_i \oplus h( (g_i \oplus h(MP_i \| d_i) ) \| T_1 )$

$= ID_i \oplus h( h( h(X_{GWN}) \oplus M_{12} ) \| T_1 )$

$\{M_1, M_2, M_3, M_{12}, T_1\}$ ⟶

### 4.2 For the sensor node $S_j$

In the registration phase, GWN computes $S_j$' s secret $x_j$ to be $h(SID_j \oplus X_{GWN} \oplus y_j)$, which is set as $h(SID_j \| X_{GWN})$ in the original scheme, where $y_j = h(X_{GWN}) \oplus r_j$ and $r_j$ is a nonce. After receiving the message from user i, $S_j$ computes $ESID_j = SID_j \oplus h(h(X_{GWN} \| 1) \| T_2) \oplus y_j$, and sends the message $\{M_1, M_2, M_3, M_{12}, T_1, T_2, ESID_j, M_4, M_5\}$ to GWN for authentication. After the above modification, we can see that even if an insider obtains a lost card and knows the parameter $e_i$; yet, from gi = $h( h(X_{GWN}) \oplus h(e_i \oplus ID_i \oplus d_i) ) \oplus h(MP_i \| d_i)$, he cannot compute the value $h(X_{GWN})$, which is now further xored by $h(e_i \oplus ID_i \oplus d_i)$ and protected in the outer hash function. Due to the one-way hash

**Fig. 5.** Modified GWN Registration phase and Sensor Node Authentication Phase

| **Sensor Node ($S_j$)** | GWN |
|---|---|
| **Registration phase** | **C**omputes<br>$x_j = h(SID_j \oplus X_{GWN} \oplus y_j)$<br>$x_j = h(SID_j \| X_{GWN})$ (original scheme)<br>$y_j = h(X_{GWN}) \oplus r_j$ |
| **Authentication phase**<br>**C**omputes<br>$ESID_j = SID_j \oplus h(h(X_{GWN} \| 1) \| T_2) \oplus y_j$<br>$\{M_1, M_2, M_3, M_{12}, T_1, T_2, ESID_j, M_4, M_5\}$ ⟶ | |

function and the unknown values of $ID_i$ and $d_i$, each user cannot obtain $h(X_{GWN})$ to launch an insider attack, because $h(X_{GWN})$ does not equal to $g_i \oplus h(MP_i \| d_i)$. Hence, the smart card loss password guessing attack is excluded. Also, he may corrupt Sj, to get $h(X_{GWN} \| 1)$; however, without the knowledge of gateway node' s secret XGWN, he cannot calculate $SID_j = ESID_j \oplus h(h(X_{GWN} \| 1) \| T_2) \oplus y_j$, where $y_j = h(X_{GWN}) \oplus r_j$. Thus, the anonymity breach is patched.

*Journal of Engineering Technology*
*Volume 8, Issue 1, Jan. 2019, PP.143-157*
RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

### 4.3 Password change phase

In addition to the above modifications, we enable a registered user $U_i$ to be able to offline change its password at will when needs by using only the smart card SC without affecting the authentication process or changing any data in the GWN or any sensor node. An illustration of the phase is depicted in Fig. 6. To change the password, $U_i$ first logins into the SC using his IDi and current $PW_i$. After SC successfully verifies $U_i$ by using the equation $e_i=h(MP_i||ID_i)$, it then proceeds with changing the password $PW_i$ to $PW_i$'. To attain this, SC must change all the values, $e_i$, $f_i$, and $g_i$, stored in the memory, including $PW_i$. For this purpose, for these values changes SC first compute values $d_i = f_i \oplus h(MP_i||e_i)$ and $h(X_{GWN}) = g_i \oplus h(MP_i \,||\, d_i)$ by using the current $e_i$, $MP_i$ and $g_i$. After that, SC computes the new $e_i$', $f_i$' and $g_i$' by using the new password $PW_i$' (i.e. $MP_i' = h(r_i \,||\, PW_i')$), replaces these to the corresponding old values in the memory, and ends up the password change phase.

**Fig.6.** password change phase of the modified scheme

Ui
Knows its $ID_i$ and $PW_i$
Has a SC = { $r_i$, $e_i$, $f_i$, $g_i$ }

User: Inserts SC into a terminal

User: Inputs $PW_i$ and $ID_i$

SC: $MP_i = h(r_i \,||\, PW_i)$

SC: $e_i = ? \; h(MP_i \,||\, ID_i)$

SC: $d_i = f_i \oplus h(MP_i||e_i)$

SC: $h(X_{GWN}) = g_i \oplus h(MP_i \,||\, d_i)$

User: Chooses and inputs new password $PW_i'$

SC: $MPi' = h(r_i \,||\, PW_i')$

SC: $e_i' = h(MP_i' \,||\, ID_i)$

SC: $f_i' = d_i \oplus h(MP_i' \,||\, e_i')$

SC: $g_i' = h(X_{GWN}) \oplus h(MP_i' \,||\, d_i)$

SC: Changes $e_i$ with $e_i'$

SC: Changes $f_i$ with $f_i'$

SC: Changes $g_i$ with $g_i'$

## 5 Security analysis

In this section, we show why our scheme can meet Liao et al.'s requirements [13] for a smart-card based password authentication protocol.

### 5.1 The user password is not stored on the server.

Our scheme requires no verifier tables on the server side. Hence, it meets the need.

### 5.2 The user can freely choose/change the password.

Since in our modification, the password change request can be accepted only after the smart card has authenticated the user. The user can they reset his password without any limitations. In other word, that our modification guarantees that only the real card holder can choose and change his password.

### 5.3 The password cannot be revealed by the administrator of the server.

The password is not revealed to the administrator of the server in either the login and authentication phase, or password change phase in our modification scheme. Thus, the modification meets this requirement.

### 5.4 The user password is not transmitted in plain form over the internet.

As shown in Section 3, the password in our scheme is not transmitted in clear form.
Hence, our scheme also satisfies this rule.

### 5.5 The scheme can resist insider attacks.

An insider attack means that a legal user J can impersonate another user U to gain the service of server S. Assume that in the modification, J wants to impersonate U to login to S; however, without the knowledge of U's password $PW_i$ and $MP_i = h(r_i \| PW_i)$, he cannot pass GWN's verification.

### 5.6 The scheme can resist the replay, password-guessing, modification-verifier-table, and stolen-verifier attacks.

Our modification can resist the modification-verifier-table attack and stolen-verifier attack, because it requires no verifier table. Meanwhile, our scheme can avoid the replay attack, because it chooses two fresh nonces, $r_i$ and $r_j$ in each protocol run. Besides, the on-line password guessing attack cannot succeed, because without the values $ID_i$, $PW_i$, $r_i$, and $r_j$, the attacker cannot compute $MP_i$ and $MP_j$ for generating the required parameters $e_i$, $d_i$, $g_i$ and $f_i$ to pass GWN's verification.

### 5.7 The length of a password is appropriate for memorization.

In our scheme, $PW_i$ is included in $MP_i = h(r_i \| PW_i)$, which is then used to generate parameters $e_i$, $d_i$, $g_i$ and $f_i$ in the message flow. Hence, our scheme's security strength doesn't rely on the length of the password. The user, thus can choose the password with any length for easy memorization.

*Journal of Engineering Technology*
*Volume 8, Issue 1, Jan. 2019, PP.143-157*
RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

### 5.8 The scheme can be efficient and practical.

Our scheme requires no complex computations. It uses hash functions and X-or operations, as does in the original scheme. Therefore, our scheme was efficient and thus practical.

### 5.9 The scheme can achieve mutual authentication.

In our scheme, both the server and the user must confirm each other's identity before generating the common session key. This means that mutual authentication should be achieved. In the following, we prove the reason why our scheme can achieve this goal.

**Mutual authentication:**
In the login and authentication phase, to confirm the user, GWN has to verify the validity of $M_3 = h(M_1 // M_2 // K_i // T_1)$, and the user must check the validity of $M_8 = h(M_6 // d_i // T_3)$ to authenticate GWN. Then, if $M_5 = h(SID_j // M_4 // T_1 // T_2 // K_j)$ , GWN confirms that $S_j$ is authentic. And if $M_{10}$ is equal to $h(SK // M_6 // M_8 // T_3 // T_4)$, $U_i$ ensures the authenticity of $S_j$. In other words, after the three parties complete the validity checks, they authenticate each other.

### 5.10 It resists against lost smart card password-guessing attacks.

When an attacker AE obtains a lost smart card, he may launch a password-guessing attack in two scenarios: (1) after $U_i$'s registration but before his login, (2) after $U_i$'s login and authentication phase. In the following, we demonstrate why our scheme can resist these two attacks.
(1) AE obtained $U_i$'s smart card after $U_i$'s registration.
Although AE can read the values ri, $e_i$, $f_i$, $g_i$ stored in the card, where $e_i = h(MP_i // ID_i)$, $f_i = d_i \oplus h(MP_i // e_i)$, $g_i = h( h(X_{GWN}) \oplus h(e_i \oplus ID_i \oplus d_i) ) \oplus h(MP_i // d_i)$, $d_i = h(ID_i // X_{GWN})$; however without di, he cannot launch such an attack. Because if he guesses the password as pwi, without IDi in ei and di in fi , he has no criteria to confirm whether his guessing is right. Thus, AE fails in this case. Even an internal legal user launch such an attack, he cannot succeed as well. Because $d_i$ does not be stored in his smart card for him to offline guess $X_{GWN}$, which then can be used to launch a password guessing attack if he obtains the other user's smart card. For example, he may try to deduce $g_i = h( h(X_{GWN}) \oplus h(e_i \oplus ID_i \oplus d_i) ) \oplus h(MP_i // d_i)$; however, he cannot succeed without the knowledges of $X_{GWN}$, and that user's di. Thus, we conclude that AE will fail when launching such as attack in this situation.
(2) AE obtained U's smart card after the login and authentication phase.
As in the former case, we can easily see that AE cannot have any advantage in deducing any helpful result in our modification scheme. Although he might intercept the two transmitted forward backward message pairs, as shown in figure 4 and 5: (a) one pair is between $U_i$ and $S_j$, and the other (b) between $S_j$ and GWN, he is not able to launch the lost smart card password guessing attack due to the same reason as in the former case. We take the values $M_1$, $M_{12}$ in case (a), and $ESID_j$, $M_9$ in case (b) as examples to demonstrate this situation. Due to that the values in (a) are: $M_1 = ID_i \oplus h( (g_i \oplus h(MP_i // d_i) ) // T_1 ) = ID_i \oplus h( h( h(X_{GWN}) \oplus M_{12} ) // T_1 )$, $M_{12} = h(e_i \oplus ID_i \oplus d_i)$, and the values in (b): $ESID_j = SID_j \oplus h(h(X_{GWN} // 1) // T_2) \oplus y_j$, where $y_j = h(X_{GWN}) \oplus r_j$ is a new set parameter in the modification, $M_9$=

$h(M_7 // x_j // T_3)$, where $M_7 = K_i \oplus h(x_j // T_3)$, $K_i = M_2 \oplus h(d_i // T_1)$, $x_j = h(SID_j \oplus X_{GWN} \oplus y_j)$, $d_i = h(ID_i // X_{GWN})$, all the four parameters mentioned ultimately contain at least one unknown value to AE; for instance, $d_i$ in $M_1$, $M_{12}$, and $y_j$ in $ESID_j$ and $M_9$. Thus, we conclude that AE will fail when launching such as attack. The others can be analyzed in the same manner. We omit them here.

## 6 Comparisons and Discussions

In this section, we first make comparisons of our modification with the state of the art, then discuss its applications in a real word and how it will be used in our future work.

### 6.1 Comparisons

We compare our scheme with several protocols in the state of tart [14-17] in terms of both the required pass number (Pass No.) and the ten security features (TSF) satisfaction by Liao et al.. We summarize it in Table 1.

**Table 1. Comparison with several protocols in the state of the art in terms of passes and STSF**

| Scheme / Attri. | [14] | [15] | [16] | [17] | Ours |
|---|---|---|---|---|---|
| Pass No. | 2 | 3 | 2 | 3 | 2 |
| TSF | x | x | x | x | o |

Symbols: x represents that the scheme cannot satisfy TSF, o an opposite to x

### 6.2 Discussions

Based on our modification, which meets Liao et al.'s ten security demands and is more secure and efficient than the other current relevant work, we can see that it is useful when applied in a real world, especially in an IOT (cloud) environment, which is prone to security loopholes and may contain more servers to cope with many users.

As the rapid development in physical material, we can image how interesting it is for this modification to be applied in a quantum identity authentication design. Therefore, in our future work, we will adapt and apply our modification to a quantum system which requires the involved parties to identify the other party through quantum channel.

## 7. Conclusion

**I**n this paper, we show that Farasha et al.'s scheme is flawed, because it suffers from (1) The smart card loss password guessing attack and (2) Anonymity breach. We have described the reasons why in Section 3. To further enhance its security, we change the messages in the registration phase and the login and authentication phase, respectively, and also let the user can change his password. From the

analysis shown in Section 4, we conclude that we have corrected the security issues in Farasha et al.'s scheme. And from Section 5, we determine that our modification meets the ten security requirements for a smart card based authentication system argued by Liao et al.. Finally, we make comparisons with the state of the art and found that our scheme is either safer or more efficient with only two passes than several of the other recent schemes.

**References**

[1] Chun-Ta Li, Min-Shiang Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards", Journal of Network and Computer Applications, Volume 33, Issue 1, January 2010, Pages 1–5

[2] Wen-Chung Kuo, Hong-Ji Wei, Jiin-Chiou Cheng, "An efficient and secure anonymous mobility network authentication scheme", journal of information security and applications 19 (2014) 18-24

[3] Jue-Sam Chou, Yalin Chen, "An Efficient Two-Pass Anonymous Identity Authentication Protocol Using a Smart Card", Jokull, Vol 63, No. 8;Aug 2013

[4] Ding Wang, Ping Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks", Ad Hoc Networks 20 (2014) 1–15

[5] Ding Wang, Nan Wang b, Ping Wang, Sihan Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity", Information Sciences 321 (2015) 162–178

[6] Muhamed Turkanovic´, Boštjan Brumen, Marko Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion", Ad Hoc Networks 20 (2014) 96–112

[7] Kaiping Xue, Peilin Hong, Changsha Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture", Journal of Computer and System Sciences 80 (2014) 195–206

[8] Ding Wang, Ping Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions" Computer Networks 73 (2014) 41–57

[9] Chun-Ta Li, Cheng-Chi Lee , "A novel user authentication and privacy preserving scheme with smart cards for wireless communications", Mathematical and Computer Modelling 55 (2012) 35–44

[10] Ding Wang, Ping Wang,"Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks", Ad Hoc Networks 20 (2014) 1–15

[11] Mohammad Sabzinejad Farasha, Muhamed Turkanovic, Saru Kumaric,Marko Hölblb,"An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment" Ad Hoc Networks 36 (2016) 152–176

[12] Celia Li, Uyen Trang Nguyen, Hoang Lan Nguyen, Nurul Huda, "Efficient authentication for fast handover in wireless mesh networks", computers & securit y 37( 2013) I 24 -I 42

[13] I-En Liao, Cheng-Chi Lee, Min-Shiang Hwang, "A password authentication scheme over insecure networks", Journal of Computer and System Sciences, Vol. 72, No. 4, pp. 727-740, 2006.

[14] Gupta, A., Tripathi, M., Shaikh, T. J., & Sharma, A., "A Lightweight Anonymous User Authentication and Key Establishment Scheme for Wearable Devices", Computer Networks, 2018.

[15] Geeta Sharma, and Sheetal Kalra. "A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications." Journal of information security and applications 42 (2018): 95-106.

[16] Dhillon, Parwinder Kaur, and Sheetal Kalra. "A lightweight biometrics based remote user authentication scheme for IoT services." Journal of Information Security and Applications 34 (2017): 255-270.

[17] Lwamo, Nassoro MR, et al. "SUAA: A Secure User Authentication Scheme with Anonymity for the Single & Multi-server Environments." Information Sciences 477 (2019): 369-385.

[18] Yalin Chen, Jue-Sam Chou, Wen-Yi Tsai "Comments on Three Multi-Server Authentication Protocols", Jokull, Vol 63, No. 7;Jul 2013

[19] Ahmed Mateen, Qing ShengZhu, Maida Seharr, "Comparative analysis of multi protocols in wireless multimedia sensor", Journal of Engineering Technology, Volume 6, Issue 2, July. 2018, PP. 292-300.

[20] A.ur-Rehman Butt, S. Ahmad, M. Asif, M. A. Javed, M. Wasim, M. H. Chaudary, M. Iqbal, "Towards an Internet of Things (IoT) based Big Data Analytics", Journal of Engineering Technology, Volume 6, Issue 2, July. 2018, PP. 70-82

[21] S. Khedekar, C. Sakarwar, M. Mukhopadhyay, "Implementation of One Time Pad based encryption: Way to unbreakable encryption and Introduction of Pseudo OTP generation", Journal of Engineering Technology, Volume 7, Special Issue (Internet of Things) Oct. 2018, PP. 62-73